Problem 5 (6 Points)

Stress-strain measurements have been collected for many samples across many parts, resulting in much noisier data than would come from a tensile test, for example. Your job is to train an ensemble of decision trees that can predict stress for an input strain.

Scikit-Learn's `RandomForestRegressor()` has several parameters that you will experiment with below.

Run each cell; then, experiment with different settings of the `RandomForestRegressor()` to answer the questions at the end.

In [3]:
```
# Import libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
%matplotlib inline
from ipywidgets import interact, interactive, fixed, interact_manual, Layout, FloatSli

# Load the data
y = np.array([133.18473289, 366.12422297, 453.70990214, 479.37136253, 238.16361712,  3
x = np.array([0.47358185, 0.80005535, 1.10968143, 1.85282726, 0.58177792, 0.24407275,
```

In [4]:
```
def plot(n_estimators, max_leaf_nodes, bootstrap):
    n_estimators = [1,10,20,30,40,50,60,70,80,90,100][int(n_estimators)]
    max_leaf_nodes = int(max_leaf_nodes)
    model = RandomForestRegressor(n_estimators=n_estimators,
                                  bootstrap=(True if "On" in bootstrap else False),
                                  max_leaf_nodes=max_leaf_nodes,
                                  random_state=0)
    model.fit(x.reshape(-1,1), y)

    xs = np.linspace(min(x),max(x),500)
    ys = model.predict(xs.reshape(-1,1))

    plt.figure(figsize=(5,3),dpi=150)
    plt.scatter(x,y,s=20,color="cornflowerblue",edgecolor="navy",label="Data")
    plt.plot(xs, ys, c="red",linewidth=2,label="Mean prediction")
    for i,dt in enumerate(model.estimators_):
        label = "Tree predictions" if i == 0 else None
        plt.plot(xs, dt.predict(xs.reshape(-1,1)), c="gray",linewidth=.5,zorder=-1, la

    plt.legend(loc="lower right",prop={"size":8})
    plt.xlabel("Strain, %")
    plt.ylabel("Stress, MPa")
    plt.title(f"Num. estimators: {n_estimators}, Max leaves = {max_leaf_nodes}, Bootst
    plt.show()

slider1 = FloatSlider(
    value=2,
    min=0,
    max=10,
    step=1,
```

```
    description='# Estimators',
    disabled=False,
    continuous_update=True,
    orientation='horizontal',
    readout=False,
    layout = Layout(width='550px')
)

slider2 = FloatSlider(
    value=5,
    min=2,
    max=25,
    step=1,
    description='Max Leaves',
    disabled=False,
    continuous_update=True,
    orientation='horizontal',
    readout=False,
    layout = Layout(width='550px')
)

dropdown = Dropdown(
    options=["On (66% of data)", "Off"],
    value="On (66% of data)",
    description='Bootstrap',
    disabled=False,
)


interactive_plot = interactive(
    plot,
    bootstrap = dropdown,
    n_estimators = slider1,
    max_leaf_nodes = slider2
    )
output = interactive_plot.children[-1]
output.layout.height = '500px'

interactive_plot
```

Out[4]:
```
interactive(children=(FloatSlider(value=2.0, description='# Estimators', layout=Layou
t(width='550px'), max=10.…
```

Questions

1. Keep bootstrapping on and set max leaf nodes constant at 3. Describe what happens to the mean prediction as the number of estimators increases.

Increasing the number of estimators leads to improved data generalization. This improvement is attributed to reduced sensitivity to outliers, resulting in a more stable and enhanced mean prediction that tends to converge. However, beyond a certain point, the performance plateaus and remains consistent.

1. Keep bootstrapping on and set number of estimators constant at 100. Describe what happens to the mean prediction as the leaf node maximum increases.

Increasing the number of leaf nodes can have the opposite effect of reducing generalization, which ultimately results in overfitting the data. Overfitting implies that the model fits the training data excessively well, indicating that the variance in the data rises, making it more responsive to outliers. Moreover, individual trees within the model may begin to overfit, contributing to a less stable mean prediction and diminishing the overall performance of the model.

1. Now disable bootstrapping. Notice that all of the predictions are the same -- the gray lines are behind the red. Why is this? (Hint: Think about the number of features in this dataset.)

Disabling bootstrapping causes predictions to stay constant due to the absence of data variation. This lack of variation results in fewer opportunities to discern patterns, ultimately leading to overfitting when the number of leaf nodes increases. Additionally, the reduction in feature randomness contributes to uniform predictions across the data.

In [ ]: